

Penentuan Ukuran *Batch* pada *Batch Processor* untuk Meminimasi *Makespan*

Filscha Nurprihatin

Program Studi Teknik Industri, Fakultas Teknologi dan Desain
Universitas Bunda Mulia
Jl. Lodan Raya No. 2, Ancol - Jakarta Utara
E-mail: fnurprihatin@bundamulia.ac.id

Received 1 March 2016; Accepted 2 May 2016

Abstract

This research determines single-item batch sizing on a batch processor with increasing batch processing time linearly if batch size is more than critical size. The set up time of each batch is constant and it is managed before raw material exist on a machine. The decision variables in this research are the number of batches and batch sizes to minimize makespan. This research assumes all orders have to be delivered to the customer at the same time (common due date). This research also develops an algorithm for common due date case studies. The resulted model development can solve the batch sizing problem and the quality of solution is feasible. For demand 125 tons and the capacity of a batch is 50 tons, then the formed batch sizes are 37, 44 and 44, or 37, 50 and 38 with the same value of makespan (427,8 minutes).

Keywords: batch sizing, single item, batch processor, makespan

1. PENDAHULUAN

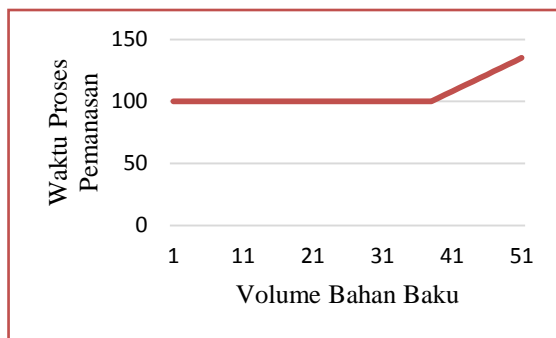
Proses pengolahan pada pabrik billet baja membutuhkan bahan baku utama berupa *scrap* dan bijih besi. Pada pabrik billet tersebut, terdapat 3 (tiga) tahap pemrosesan, yaitu proses pemanasan (*steelmaking*) pada bagian *Electric Arc Furnace (EAF)*, proses penambahan elemen *ferroalloys* pada bagian *Ladle Furnace (LF)*, dan proses pencetakan pada bagian *casting*. Proses pemanasan membutuhkan waktu terpanjang di antara ketiga tahap pengolahan tersebut. Oleh karena itu, pengurangan *makespan* pada proses pemanasan tersebut mempunyai pengaruh yang signifikan terhadap pengurangan total waktu pengolahan billet baja. Bashori (2015) menyatakan *makespan* merupakan waktu yang diperlukan untuk menyelesaikan seluruh *batch* pada *shop*.

Pada proses pemanasan di bagian *Electric Arc Furnace (EAF)* bahan baku ditempatkan pada wadah yang disebut *ladle* dengan kapasitas 50 ton. Proses pemanasan dilakukan sampai suhu bahan baku tersebut mencapai 1620°C dengan waktu proses selama 100-135 menit. Rentang waktu proses ini tergantung pada volume bahan baku yang diproses. Bila volume *ladle* kurang dari atau

sama dengan ukuran kritis (37 ton), maka waktu proses yang dibutuhkan adalah 100 menit, tetapi bila *ladle* penuh (volume maksimal), maka waktu proses yang dibutuhkan adalah 135 menit. Waktu proses meningkat secara linear bila volume bahan baku lebih dari ukuran kritis (37 ton) sampai dengan kapasitas *ladle*. Setiap penambahan 1 (satu) ton bahan baku, maka waktu proses pemanasan akan meningkat kira-kira $(135-100)/(50-37) = 2,7$ menit. Ilustrasi hubungan volume bahan baku terhadap waktu proses dapat dilihat pada Gambar 1.

Pada tahap *batching*, ukuran *batch* yang terbentuk bisa lebih besar dan lebih kecil dari ukuran kritis. Dengan ukuran *batch* lebih besar dari ukuran kritis, maka waktu proses *batch* meningkat, sementara bila ukuran *batch* lebih kecil dari ukuran kritis, maka terdapat kapasitas menganggur yang dapat digunakan untuk pemrosesan *batch* dengan waktu proses konstan. *Set up* pada mesin dilakukan ketika mesin akan memproduksi *batch*, sehingga jumlah *batch* akan mempengaruhi jumlah *set up*. Semakin banyak jumlah *batch* (semakin kecil ukuran *batch*), maka akan semakin banyak jumlah *set up* yang dilakukan. Demikian juga sebaliknya,

semakin sedikit jumlah *batch* (semakin besar ukuran *batch*), maka akan semakin sedikit jumlah *set up* yang dilakukan.



Gambar 1. Grafik Hubungan Volume Bahan Baku terhadap Waktu Proses Pemanasan

Penelitian dengan ukuran performansi minimasi *makespan* dilakukan oleh Oyetunji dan Oluleye (2011), Haddad dan Moradinezhad (2012) dan Bashori (2015). Oyetunji dan Oluleye (2011) mempertimbangkan faktor waktu rilis untuk meminimasi *makespan* pada mesin tunggal dengan menggunakan algoritma usulan yang dibandingkan dengan algoritma *branch and bound* dan algoritma AEO. Haddad dan Moradinezhad (2012) melakukan penelitian pada mesin majemuk yang terdeteriorasi menggunakan meta heuristik. Bashori (2015) menerapkan algoritma *cross entropy* pada proses produksi *flow shop*.

Studi tentang penjadwalan *batch* untuk *single item* pada mesin tunggal dilakukan oleh Halim *et al.* (1994) dengan sistem *batch* berupa *job processor*, yaitu mesin yang memproduksi *part* dalam *batch* satu persatu. Artinya, waktu yang dibutuhkan oleh *batch* adalah penjumlahan dari waktu proses semua *part* yang terdapat di dalam *batch* tersebut. Ukuran performansi yang digunakan adalah minimasi total waktu tinggal aktual *part* dalam *batch* yang didefinisikan sebagai waktu antara saat datang *batch* dengan *due date* dikalikan dengan jumlah *part* dalam *batch* tersebut.

Penelitian Hidayat *et al.* (2013) membahas masalah penjadwalan pada *batch processor*. Pada *batch processor*, seluruh *part* dalam *batch* diproses secara bersamaan. Artinya, waktu proses *batch* bukan jumlah waktu proses semua *part* yang terdapat di dalam *batch* tersebut, tetapi sama dengan waktu proses sebuah *part*. Dalam penelitian Hidayat *et al.* (2013) waktu proses *batch* dianggap konstan dan tidak dipengaruhi oleh ukuran *batch*. Ukuran performansi yang digunakan adalah minimasi total waktu tinggal aktual *part*. Penelitian Hidayat *et al.* (2013) juga merumuskan total waktu tinggal aktual *batch* untuk kasus *single item common due date*.

Penelitian penjadwalan pada *batch processor* dilakukan oleh Astuti (2014) dengan waktu proses setiap *batch* tidak konstan. Waktu proses *batch* yang tidak konstan tergantung pada waktu pemasangan *tools* untuk seluruh *part* dalam *batch*. Semakin banyak *part* dalam suatu *batch*, maka akan semakin panjang waktu pemasangan *tools* yang diperlukan. Ukuran performansi yang digunakan adalah minimasi total waktu tinggal aktual *part*. Penelitian Astuti (2014) membahas kasus *single item common due date* dengan waktu proses yang meningkat tergantung ukuran *batch*, sedangkan waktu proses dalam penelitian ini meningkat bila ukuran *batch* lebih besar dari ukuran kritis (L).

Penelitian pada *batch processor* dengan ukuran performansi meminimasi *makespan* dilakukan oleh Damodaran dan Santiago (2011), Li (2012), Sahreian *et al.* (2012), Ozturk *et al.* (2012), Zhu *et al.* (2014) dan Trindade *et al.* (2014). Damodaran dan Santiago (2011) meneliti *multi batch processor* pada sistem produksi *job shop* dengan menggunakan formulasi matematis. Li (2012) melakukan penelitian pada *batch processor* yang disusun secara paralel dengan mempertimbangkan *release times* dan ukuran *job*. Pada penelitian Sahreian *et al.* (2012) waktu proses *job*, waktu *set up* dan waktu rilis bersifat stokastik. Zhu *et al.* (2014) membahas *multi batch processor* yang disusun secara paralel. Trindade *et al.* (2014) membahas *batch processor* tunggal dengan waktu rilis dan ukuran *job* yang berbeda dengan menggunakan *mixed integer linear programming*. Waktu proses *batch* pada penelitian-penelitian tersebut adalah waktu terpanjang dari *job* pada *batch*.

Ozturk *et al.*, (2012) membahas penelitian pada *multibatch processor* dengan yang disusun secara paralel dengan waktu rilis dan ukuran *job* yang berbeda. Waktu proses *batch* adalah tetap dan tidak tergantung oleh ukuran *batch*.

Dengan demikian dapat dilihat bahwa perlu dilakukan penelitian ke arah pengembangan model penjadwalan pada *batch processor* dengan waktu proses *batch* meningkat bila ukuran *batch* lebih dari ukuran kritis (L). Penelitian ini dilakukan dengan menggunakan kriteria performansi *makespan*. Pada setiap pemrosesan, *batch processor* yang digunakan untuk memproses bahan baku membutuhkan waktu *set up* (s) yang konstan dan dilakukan sebelum kedatangan *part* di lantai produksi. Terdapat *trade off* antara jumlah *batch* dan jumlah *set up* yang dilakukan. Bila ukuran *batch* kecil, maka jumlah *batch* yang terbentuk dan jumlah *set up* menjadi besar, sehingga bisa membuat *makespan* besar. Sebaliknya, bila ukuran *batch* besar, maka jumlah *batch* yang terbentuk dan jumlah *set up* menjadi kecil, tetapi waktu proses

batch menjadi besar karena ukuran batch lebih besar dari ukuran kritis (L). Kondisi ini juga bisa meningkatkan makespan.

2. METODOLOGI

2.1 Pengembangan Model

Pada tahap ini dilakukan pemilihan model acuan ini didasarkan atas kesamaan taksonomi penelitian dari masing-masing penelitian yang relevan dengan masalah yang dihadapi sesuai dengan tujuan penelitian. Pada Gambar 2 dapat dilihat pengembangan model dilakukan dengan mengacu pada tiga penelitian terdahulu yaitu:

1. Ozturk *et al.* (2012) membahas model penjadwalan pada multi batch processor yang disusun secara paralel. Penelitian Hidayat *et al.*, (2013) membahas model penjadwalan pada batch processor tunggal dengan waktu proses batch konstan. Waktu proses batch yang konstan pada penelitian Ozturk *et al.* (2012) dan Hidayat *et al.* (2012) menjadi acuan dalam memformulasikan waktu proses batch yang konstan jika ukuran batch lebih kecil atau sama dengan ukuran kritis.
2. Astuti (2014) mengembangkan model penjadwalan pada batch processor tunggal dengan waktu proses batch tidak konstan. Waktu proses yang tidak konstan dalam penelitian ini menjadi acuan dalam memformulasikan penambahan waktu proses untuk ukuran batch lebih dari ukuran kritis.
3. Penelitian Trindade *et al.* (2014) memformulasikan model penjadwalan pada batch processor tunggal dengan waktu proses batch tidak konstan untuk meminimasi makespan. Formulasi fungsi tujuan minimasi

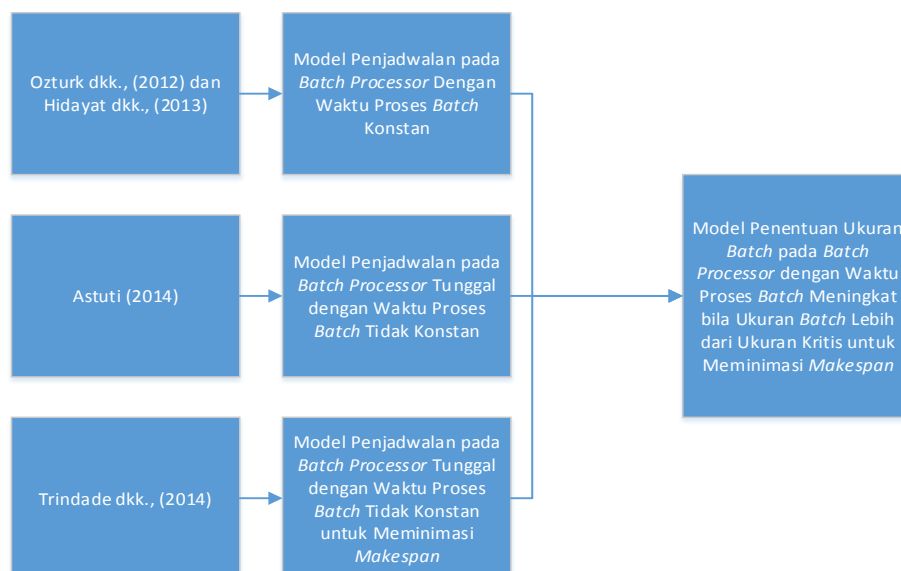
makespan dalam Trindade *et al.* (2014) menjadi acuan dalam merumuskan fungsi tujuan penelitian ini.

Untuk menguji model ini, maka dilakukan uji numerik yang akan ditampilkan dalam Hasil dan Pembahasan.

2.2 Formulasi Masalah

Dengan menggunakan variabel-variabel yang telah ditentukan, dilakukan pengembangan model penentuan ukuran batch pada batch processor untuk single item dengan waktu proses batch meningkat bila ukuran batch lebih dari ukuran kritis untuk meminimasi makespan. Model referensi yang dijadikan acuan tidak bisa langsung digunakan untuk memecahkan masalah, karena itu diperlukan pengembangan lebih lanjut untuk mendekati kondisi sistem.

Misalkan terdapat permintaan (D) yang merupakan item sejenis (single item) yang akan diproses pada batch processor tunggal. Waktu proses batch terdiri dari waktu proses minimal ($t_{Q \leq L}$) dan waktu penambahan tergantung ukuran batch ($t_{Q \leq L} + P_i$). Waktu proses untuk setiap batch (t_i) tergantung pada ukuran batch (Q_i). Pada saat $Q_i \leq L$, maka $t_i = t_{Q \leq L}$. Pada saat $Q_i > L$, maka waktu proses akan naik secara linear hingga titik maksimal kapasitas batch. Batch processor yang digunakan untuk memproses part membutuhkan waktu set up (s) pada setiap pemrosesan batch. Lamanya waktu set up untuk setiap batch tidak tergantung kepada ukuran batch. Proses set up dilakukan sebelum kedatangan batch di rantai produksi. Permasalahan disini adalah bagaimana memaksimalkan jumlah batch dengan ukuran batch \leq ukuran kritis agar makespan minimal.



Gambar 2. Tahapan Pengembangan Model

2.3 Kriteria Performansi

Kriteria performansi dalam penelitian ini adalah *Makespan*. $Makespan(C_{max})$ merupakan waktu yang diperlukan untuk menyelesaikan seluruh *batch* pada *shop*. Trindade *et al.* (2014) merumuskan fungsi tujuan minimasi *makespan* sebagai berikut:

$$\text{Min } C_{max} = S_k + P_k \quad (1)$$

dengan:

S_k : waktu mulai *batch* ke- k
 P_k : waktu proses *batch* ke- k

Fungsi tujuan ini tidak dapat langsung digunakan karena waktu proses *batch* pada Trindade *et al.* (2014) tidak konstan. Oleh karena itu, dikembangkan formulasi minimasi *makespan* untuk kasus waktu proses *batch* meningkat bila ukuran *batch* lebih besar dari ukuran kritis, sebagai berikut:

$$\text{Min } C_{max} = \sum_{i=1}^N (t_{Q \leq L} + P_i + s_i) \quad (2)$$

dengan:

i = indeks *batch* ($i = 1, 2, \dots, N$)
 P = penambahan waktu proses *batch*
 s_i = waktu *set up* pada *batch* b_i (menit)
 $t_{Q \leq L}$ = waktu proses yang dibutuhkan *batch* b_L

2.4 Notasi Matematis

Notasi matematis yang digunakan dalam penelitian ini adalah sebagai berikut:

b_i = *batch* yang berada di posisi ke- i pada *shop*
 b_L = *batch* yang berukuran kurang dari atau sama dengan L
 b_p = *batch* yang berukuran lebih dari L
 D'_i = jumlah \widehat{D} yang belum diproduksi
 i = indeks *batch* ($i = 1, 2, \dots, N$)
 L = ukuran kritis

$$N_L = \underset{N_L}{\operatorname{argmax}} \left\{ \left(\frac{\widehat{D} - N_L \cdot L}{c} \right) - N + N_L \leq 0 \right\}$$

$$Q_{Li} = \begin{cases} L; \forall i = 1, 2, \dots, N_L - 1 \\ L; D'_i = \widehat{D} - \left(\sum_{u=1}^i (Q_{L i-u}) \right) \geq L; \forall i = N_L \\ D'_i; D'_i = \widehat{D} - \left(\sum_{u=1}^i (Q_{L i-u}) \right) < L; \forall i = N_L \end{cases} \quad (5)$$

$$N_p = N - N_L \quad (6)$$

P = penambahan waktu proses *batch*

2.5 Variabel Keputusan

Variabel keputusan yang akan dicari untuk meminimumkan kriteria performansi adalah sebagai berikut:

N = jumlah *batch* pada *shop*
 N_L = jumlah *batch* b_L pada *shop*
 N_p = jumlah *batch* b_p pada *shop*
 Q_i = ukuran setiap *batch* b_i
 Q_L = ukuran setiap *batch* b_L
 Q_p = ukuran *batch* b_p

2.6 Parameter

Parameter-parameter yang terlibat dalam model penentuan ukuran *batch* pada mesin tunggal berproses *batch* dengan maksimasi *batch* b_L adalah sebagai berikut:

A = penambahan waktu proses setiap penambahan 1 ton bahan baku
 c = kapasitas *batch processor*
 D = jumlah permintaan (unit)
 \widehat{D} = jumlah bahan baku yang dibutuhkan (ton)
 L = ukuran kritis
 s_i = waktu *set up* pada *batch* b_i (menit)
 $t_{Q \leq L}$ = waktu proses yang dibutuhkan *batch* b_L

Jumlah *batch* minimal dapat dihitung dengan rumus berikut:

$$N = (\widehat{D}/c) \quad (3)$$

Karena waktu proses yang dibutuhkan *batch* b_L adalah minimal, maka diperlukan maksimasi *batch* b_L dengan menggunakan persamaan (4) dan ukuran *batch* b_L dijelaskan pada persamaan (5).

Sedangkan untuk menentukan jumlah *batch* b_p , menggunakan persamaan (6). Penentuan ukuran *batch* b_p (Q_{pi}) dijelaskan pada persamaan (7). Persamaan (8) menjumlahkan ukuran *batch* b_L dan *batch* b_p .

$$\forall N_L = 0, 1, 2, \dots, N \quad (4)$$

$$Q_{pi} = \begin{cases} c; D'_i = \hat{D} - \left(\sum_{i=1}^{N_L} (Q_{Li}) \right) - \left(\sum_{u=1}^i (Q_{pi-u}) \right) \geq c; \forall i = 1, 2, \dots, N_p \\ D'_i; \text{otherwise}; \forall i = 1, 2, \dots, N_p \end{cases} \quad (7)$$

$$Q_i = \sum_{i=1}^N (Q_{pi} + Q_{Li}) \quad (8)$$

2.7 Formulasi Model

Secara lengkap, model penentuan ukuran *batch* pada *batch processor* dengan waktu proses *batch*

meningkat bila ukuran *batch* lebih dari ukuran kritis sebagai berikut:

Fungsi Tujuan

$$\text{Min } C_{max} = \sum_{i=1}^N (t_{Q \leq L_i} + P_i + s_i) \quad (9)$$

Fungsi Pembatas

$$\sum_{i=1}^N Q_i = \hat{D} \quad (10)$$

$$P_i = \begin{cases} 0; \forall Q_i \leq L; \forall i = 1, 2, \dots, N_L \\ A (Q_i - L); \text{otherwise}; \forall i = 1, 2, \dots, N_p \end{cases} \quad (11)$$

$$\begin{matrix} N \\ \geq 1 \end{matrix} \quad \forall i = 1, 2, \dots, N \text{ dan integer} \quad (12)$$

$$\begin{matrix} Q_i \\ \geq 0 \end{matrix} \quad \forall i = 1, 2, \dots, N \text{ dan integer} \quad (13)$$

Fungsi (9) menyatakan tujuan model yaitu minimasi *makespanbatch* yang akan diproses. Pembatas (10) menjamin bahwa ukuran *batch* yang diproduksi sama dengan jumlah permintaan. Pembatas (11) menyatakan bahwa nilai *P* sama dengan 0 atau bernilai tertentu sesuai ukuran *batch*. Pembatas (12) menjamin bahwa jumlah *batch* pada *shop* lebih dari 1 dan merupakan bilangan integer. Pembatas (13) menjamin ukuran *batch* lebih dari 1 dan merupakan bilangan integer.

Algoritma:

- Langkah 0. Tetapkan parameter $D, s, A, L, c, t_{Q \leq L}$. Lanjut ke langkah 1.
- Langkah 1. Hitung kebutuhan baja cair (\hat{D}). Lanjut ke langkah 2.
- Langkah 2. Hitung jumlah *batch* minimal dengan menggunakan persamaan (2). Bulatkan angka yang dihasilkan ke atas. Lanjutkan ke langkah 3.
- Langkah 3. Tentukan N_L dan N_p dengan menggunakan persamaan (3) dan (5). Lanjutkan ke langkah 4.

Langkah 4.

Set $Q_{L0} = 0$ dan $Q_{P0} = 0$. Tentukan ukuran *batch* $b_L (Q_{Li})$ dan ukuran *batch* $b_P (Q_{Pi})$ dengan menggunakan persamaan (4) dan (6). Lanjutkan ke langkah 5. Hitung penjumlahan Q_{Li} dan Q_{Pi} dengan persamaan (7).

Langkah 5.

Tentukan penambahan waktu proses masing-masing *batch* (P_i) dengan persamaan (10). Kemudian tentukan waktu proses masing-masing *batch*. Lanjutkan ke langkah 6.

Langkah 6.

Urutkan *batch* dalam urutan acak. Hitung C_{max} dengan menggunakan persamaan (8). Lanjutkan ke langkah 7.

Langkah 7.

Berhenti.

3. HASIL DAN PEMBAHASAN

Hasil pengujian numerik pada model ini disajikan dalam Tabel 1. Pada kondisi 1 dan 2 terdapat *demand* yang sama, yaitu 125 ton, tetapi dengan ukuran *batch* b_p yang berbeda. Pada kondisi 1, *batch* yang terbentuk masing-masing berukuran 37, 44 dan 44 ton, sehingga waktu proses *batch* b_p yang terbentuk adalah 100 menit + 18,9 menit = 118,9 menit. Terdapat 3 (tiga) kombinasi acak urutan *batch* yang menghasilkan nilai *makespan* (C_{max}) yang sama, yaitu 427,8 menit.

Pada kondisi 2, *batch* yang terbentuk masing-masing berukuran 37, 50 dan 38 ton, sehingga waktu proses *batch* b_p dengan ukuran 50 ton adalah 100 menit + 35,1 menit = 135,1 menit. Untuk *batch* yang berukuran 38 ton, maka waktu prosesnya adalah 100 menit + 2,7 menit = 102,7 menit. Pada kondisi 2 juga terdapat 3 (tiga) kombinasi urutan *batch* yang diproduksi yang menghasilkan *makespan* yang sama, yaitu 427,8 menit. Artinya,

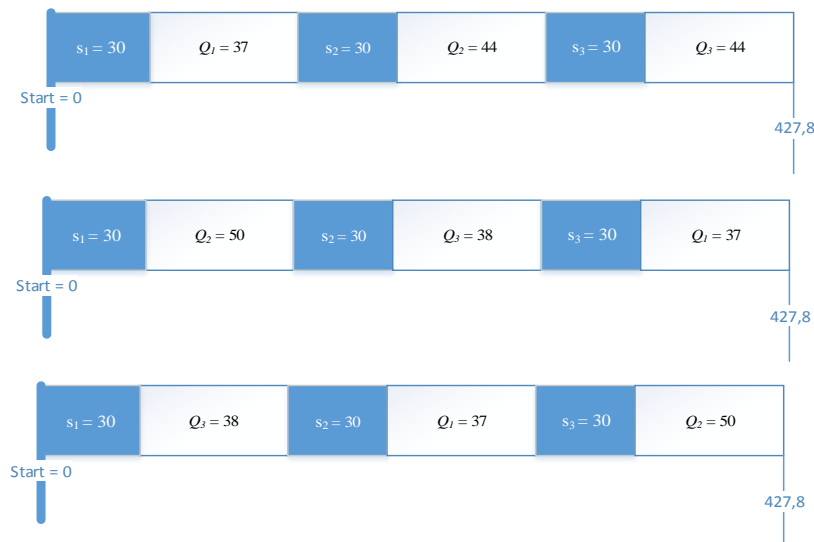
dalam kasus penelitian ini urutan *batch* tidak berpengaruh terhadap nilai *makespan*.

Pada Tabel 1, terlihat bahwa maksimasi jumlah *batch* b_L menurunkan *makespan* dan ukuran *batch* b_p tidak berpengaruh terhadap nilai *makespan*. Perubahan nilai *demand* berpengaruh pada pembentukan jumlah dan ukuran *batch*. Oleh karena itu, penambahan waktu proses juga berubah. Semakin besar *demand*, maka akan semakin besar ukuran *batch* yang terbentuk. Sebaliknya, semakin kecil jumlah *demand*, maka akan semakin kecil ukuran *batch*. Urutan pengerjaan *batch* tidak berpengaruh terhadap nilai *makespan*.

Dengan asumsi tanpa penalti ketika *item* diserahkan kepada konsumen pada saat *item* jadi setelah *due date*, maka jadwal keseluruhan yang diperoleh akan layak.

Tabel 1. Hasil Pengujian pada Model

Kondisi	<i>Demand</i> (\hat{D})	Ukuran <i>Batch</i>	Waktu Proses minimal ($t_{Q \leq L}$)	Waktu Proses Tambahan (P)	Urutan Proses	C_{max}
1	125	$Q_1 = 37$	100	0	1-2-3 1-3-2 2-1-3	427.8
		$Q_2 = 44$	100	18,9		
		$Q_3 = 44$	100	18,9		
2	125	$Q_1 = 50$	100	35.1	2-3-1 3-1-2 3-2-1	427.8
		$Q_2 = 38$	100	18,9		
		$Q_3 = 37$	100	0		
3	150	$Q_1 = 50$	100	35.1	1-2-3 1-3-2 2-1-3	495
		$Q_2 = 50$	100	35.1		
		$Q_3 = 50$	100	35.1		
4	111	$Q_1 = 37$	100	0	2-3-1 3-1-2 3-2-1	390
		$Q_2 = 37$	100	0		
		$Q_3 = 37$	100	0		



Gambar 3. Gantt Chart untuk contoh demand = 125

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Model yang dikembangkan mampu menyelesaikan permasalahan penentuan ukuran batch pada batch processor untuk single item dengan waktu proses batch meningkat bila ukuran batch lebih dari ukuran kritis. Model dapat digunakan untuk mencari solusi berupa jumlah dan ukuran batch dengan kriteria performansi meminimasi makespan.
2. Solusi jumlah dan ukuran batch untuk kasus common due date adalah feasible.
3. Maksimasi jumlah batch b_L menurunkan makespan dan ukuran batch b_P tidak berpengaruh terhadap nilai makespan.
4. Semakin besar demand, maka akan semakin besar ukuran batch yang terbentuk. Sebaliknya, semakin kecil jumlah demand, maka akan semakin kecil ukuran batch. Urutan pengerjaan batch tidak berpengaruh terhadap nilai makespan.

5. DAFTAR PUSTAKA

1. Astuti, Murni Dwi. (2014). *Model Penjadwalan Batch pada Batch Processor Tunggal dengan Waktu Proses Batch Tidak Konstan untuk Meminimasi Total Waktu Tinggal Aktual*. Tesis Program Studi Magister Teknik dan Manajemen Industri, Fakultas Teknologi Industri, Institut Teknologi Bandung.
2. Bashori, Hasan. (2015). *Upaya Meminimasi Makespan dengan Penerapan Algoritma Cross Entropy pada Penjadwalan Flow Shop*. Widya Teknika Vol.23 No.1; Maret 2015 ISSN 1411 – 0660: 10-14
3. Damodaran, Purushothaman dan Santiago, Miguel Rojas. (2011). *Mathematical Formulation to Minimize Makespan in A Job Shop with A Batch Processing Machine*. Ninth Latin American and Caribbean Conference for Engineering and Technology, August 2011, Medellin, Colombia.
4. Haddad, Hamidreza dan Moradinezhad, Dariush. (2012). *Minimizing Makespan in Parallel Scheduling Problem with Deterioration using Two Meta Heuristics*. International Journal of Science and Technology. Vol. 2, No. 6, June 2012. ISSN 2224-3577.
5. Halim, A.H., Miyazaki, S., dan Ohta, H. (1994). *Batch-scheduling Problems to Minimize Actual Flow times of Parts Through the Shop under JIT Environment*. European Journal of Operational Research, 72, 529-544.
6. Idayat, Nita P.A., Carkavastia, Andi., Samadhi, TMA Ari., Halim, Abdul Hakim. (2013). *A Single Item Batch Scheduling on a Batch Processor to Minimize Total Actual Flow Time of Parts Through the Shop*. Asia Pacific Industrial Engineering and Management System.
7. Li, Shuguang. (2012). *Makespan Minimization on Parallel Batch Processing Machines with Release Times and Job Sizes*. Journal of Software, Vol. 7, No. 6, June 2012.
8. Oyetunji, E. O. dan Oluleye, A. E. (2011). *Minimizing Makespan on Single Machine with Release Dates*. International Journal of Advancements in Technology, Vol. 2, No. 1, January 2011. ISSN: 0976-4860.
9. Sahraeian, R., Samaei, F., dan Rastgar, I. (2012). *Minimizing the Makespan on Parallel Batch Scheduling with Stochastic Times*. 6th International Conference on Industrial Engineering and Industrial Management, Vigo, July 2012.
10. Trindade, Renan Spencer., Araujo, Olinto Cesar Bassi de., Muller, Felipe Martins., Fampa,

- Marcia Helena Costa. (2014). *A New Mixed Integer Linear Programming Model for Minimizing Makespan on A Single Batch Processing Machine with Release Times and Non-Identical Job Sizes*. Simposio Brasileiro de Pesquisa Operacional.
11. Zhu, Yuqing., Jiang, Yiwei., Wu, Weili., Ding, Ling., Teredesai, Ankur., Li, Deying., Lee, Wonjun. (2014). *Minimizing Makespan and Total Completion Time in MapReduce-like Systems*. IEEE Conference on Computer Communications.